No. 23-2265

United States Court of Appeals for the Federal Circuit

EXPRESS MOBILE, INC.,

Plaintiff-Appellant,

v.

GODADDY.COM, LLC,

Defendant-Appellee.

Appeal from the United States District Court for the District of Delaware
in C.A. No. 19-cv-01937, United States District Court Judge Matthew F. Kennelly

REPLY BRIEF OF PLAINTIFF-APPELLANT

James R. Nuttall
(Lead Counsel)
jnuttall@steptoe.com
Katherine H. Tellez
Robert F. Kappers
Candice J. Kwark
STEPTOE LLP
227 West Monroe Street, Suite 4700
Chicago, IL 60606
Telephone: (312) 577-1300
Facsimile: (312) 577-1370

Counsel for Plaintiff-Appellant,
Express Mobile, Inc.

March 1, 2024

TABLE OF CONTENTS

I. ’397 PATENT FAMILY: THE DISTRICT COURT ERRED IN GRANTING SUMMARY JUDGMENT OF NON-INFRINGEMENT BASED ON AN IMPROPER CLAIM CONSTRUCTION 2

A. GD mischaracterizes XMO’s infringement theory and the claim construction and summary judgment issues before the Court. 3

B. The claim term, RTE, encompasses files that call on an API to retrieve information from a database. 8

1. The district court’s construction of RTE, by its terms, includes files that “read information from a database” through an API. ..8

2. The Court should alternatively construe RTE to require “utilizing information from the database.” 10

C. There are genuine disputes of material fact that make summary judgment inappropriate. 13

1. Under the district court’s claim construction, there are genuine disputes as to whether “script.js” uses “fetch()” to “*read information from the database*.” 13

a. There is a genuine dispute as to whether “script.js” contains the function call, “fetch().” 13

b. There is a genuine dispute as to whether “script.js” uses “fetch()” to “*read information from the database*.” 14

c. There is a genuine dispute as to whether the information read by “script.js” (via “fetch()”) comes from GD’s Cassandra database. 15

2. Under XMO’s proposed claim construction, there is a genuine dispute as to whether “script.js” “*utilizes information from the database*.” 17

II. ’755 PATENT FAMILY: THE DISTRICT COURT ERRED IN DENYING XMO’S MOTION FOR A NEW TRIAL AND JUDGMENT AS A MATTER OF LAW 18

A. XMO was entitled to a new trial because GD contradicted the construction of "device-specific code." 18

1. GD's expert improperly equated "device-specific code" with code specific to an "operating system." 19

2. GD's expert also improperly excluded web browsers from the scope of the term "platform." 24

B. The district court also erred in denying XMO's motion for JMOL....27

III. THE COURT SHOULD DISREGARD GD'S ARGUMENT ABOUT WILLFUL INFRINGEMENT 29

TABLE OF AUTHORITIES

Page(s)

Cases

Absolute Software, Inc. v. Stealth Signal, Inc.,
659 F.3d 1121 (Fed. Cir. 2011) 13

Aug. Tech. Corp. v. Camtek, Ltd.,
542 F. App'x 985 (Fed. Cir. 2013) 30

Eon Corp. IP Holdings v. Silver Spring Networks,
815 F.3d 1314 (Fed. Cir. 2016) 27

Exergen Corp. v. Wal-Mart Stores, Inc.,
575 F.3d 1312 (Fed. Cir. 2009) 20, 21

Facebook, Inc. v. Express Mobile, Inc.,
IPR2021-01126, Paper 19 (P.T.A.B. May 6, 2022) 11

Liebel–Flarsheim Co. v. Medrad, Inc.,
358 F.3d 898 (Fed. Cir. 2004) 13

Lifestyle Enters., Inc. v. United States,
751 F.3d 1371 (Fed. Cir. 2014) 24

Moba, B.V. v. Diamond Automation, Inc.,
325 F.3d 1306 (Fed. Cir. 2003) 20, 27

NobelBiz, Inc. v. Global Connect, L.L.C.,
701 F. App'x 994 (Fed. Cir. 2017) 27

Shopify Inc. v. Express Mobile, Inc.,
CA No. 19-439-RGA, 2020 WL 3432531 (D. Del. June 23, 2020).....19

Tandon Corp. v. ITC,
831 F.2d 1017 (Fed. Cir. 1987) 23

Verizon Servs. Corp. v. Cox Fibernet Va., Inc.,
602 F.3d 1325 (Fed. Cir. 2010) 22

WesternGeco LLC v. ION Geophysical Corp.,
913 F.3d 1067 (Fed. Cir. 2019) 22

Wi-LAN, Inc. v. Ericsson, Inc.,
675 F. App'x 984 (Fed. Cir. 2017) 27

Other Authorities

MDN Web Docs, *Fetch API*, https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API (last visited Feb. 28, 2024) 4

GD tries to evade appellate scrutiny by hand-waving. It spends the lion's share of its Response Brief mischaracterizing XMO's theory of infringement with respect to the '397 Patent Family, arguing that XMO relies on "dozens" of files ("a pipeline of files") to read information from a database. Resp. at 31. None of that is true. XMO is relying on a single file: "script.js." XMO's theory is that, when a browser executes "script.js," it triggers a function call aptly named "fetch()" (contained in "script.js") to call on an API and retrieve information from GD's database. With XMO's theory properly framed, the issues before the Court are decidedly clear. As for claim construction, at a minimum, the Court should confirm that "run time engine" ("RTE") includes files that call on an API to retrieve information from a database. Anything less would run counter to every expert in this case, even GD's, who testified that "using an API to access a database" is "the only way that – the only practical way." Appx11118 at 115:8–15. Then, the Court should reverse summary judgment of non-infringement because a reasonable jury could find that "script.js" is the claimed RTE properly construed.

As to the '755 Patent Family, GD fails to counter XMO's substantive arguments that it contradicted the construction of the claimed player, which requires "code that is specific to the operating system, programming language, or platform of a device." In particular, GD's trial presentation repeatedly and improperly equates "device-specific code" with code specific to an operating system and

excludes web browsers. Instead of countering those arguments, GD attempts to justify its actions on the basis that it also blandly recited the full, correct construction during trial. Any such recitation does nothing to cure the repeated contradictions. *E.g.*, Appx17454 at 628:13 (insisting that the term "essentially" just refers to the "operating system"). And importantly here, GD cannot save its improper statements of scope at least because the district court below, in adopting the prior reasoning of the *Shopify* summary judgment decision, had already acknowledged that browser-specialized code met the requirement of code specific to a "platform." Appx056. Ultimately, a new trial, if not JMOL of infringement, is warranted in view of the degree of contradiction that infected the trial.

I. '397 PATENT FAMILY: THE DISTRICT COURT ERRED IN GRANTING SUMMARY JUDGMENT OF NON-INFRINGEMENT BASED ON AN IMPROPER CLAIM CONSTRUCTION

The district court erred in granting summary judgment of non-infringement of the '397 Patent Family. GD mischaracterizes XMO's infringement theory in an effort to confuse the straightforward issues before the Court. As to claim construction, the Court should confirm that the relevant term, RTE, includes files that retrieve information from a database by calling on an API (which all experts agree is how files read information from databases). With the issues properly framed, it is clear that genuine disputes of material fact should have precluded summary judgment. The Court should reverse.

A. GD mischaracterizes XMO's infringement theory and the claim construction and summary judgment issues before the Court.

GD misdirects the Court about XMO's infringement theory to confuse the issues that the Court must actually decide. *Infra* §§ I.B. and I.C. XMO does not rely on "dozens of intermediary files and disparate programs" ("a pipeline of files") to read information from a database. Resp. at 31. XMO's theory of infringement has always been that GD's JavaScript file, "script.js," contains the function call, "fetch()," which is used to read information from GD's Cassandra database. It does so through the undisputedly most-common method—by calling on an API for the database called "DPS-API" (which has the codes to access the database). Appx15195.

XMO identified "script.js" as an RTE. In his Opening Report, Dr. Almeroth identified "script.js" as a "third category of run time files." Appx5418; Appx6004–6009.

"script.js" contains the function call, "fetch()." Dr. Almeroth excerpted code from "script.js" and identified its function call, "fetch()," as responsible for "read[ing] information from databases on the server." Appx6006 (excerpting code and highlighting "fetch()"). "fetch()," in fact, is a well-known function call in JavaScript used to download information from a database over a network.

Appx6006; Appx10578.[1]

When a browser executes "script.js," "fetch()" requests information from GD's Cassandra database by calling on DPS-API. Dr. Almeroth explained how "fetch()" requests information from GD's Cassandra database by calling on "DPS-API." Appx10577–10578. GD does not dispute that "DPS-API" has database access codes (the "Priam" library of codes) to access the Cassandra database. Appx5427; Appx4794 (describing "the open source 'Priam' library of codes (that contain the Cassandra database access codes)"); Appx4796 (depicting "DPS-API" as including the "DB access codes"); Appx4795 ("DPS-API . . . can read Cassandra via the Priam database access"); *see also* Appx6018; Appx6021. GD likewise does not dispute that using an API is the most-common, if not the only, way to read from a database. Appx11118 at 115:8–15; Appx11540-11541 at 169:8–170:11.

GD admits that settings selected by its users (e.g., colors, fontScale, etc.) are stored on GD's Cassandra database. Appx5362–5363; Appx5988–5989; *see also* Appx5984–5993; Appx4794; Appx14764–14765 at 97:1–98:7; Appx17445 at 591:10–13. "fetch()" returns those user-selected settings to "script.js," which then uses that information to display a webpage or website. Appx6005–6007. For example, "script.js" assigns user-selected colors and fontScale settings (retrieved

---

[1] *See also* MDN Web Docs, *Fetch API*, https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API (last visited Feb. 28, 2024).

from the Cassandra database by "fetch()") to a data structure called "window.wsb" (wsb: web site builder), as shown in excerpts from Dr. Almeroth's reports. Appx6005; *see also* Appx6006–6007; Appx5422. In this regard, "script.js" dynamically builds a display of the webpage when executed by a browser.

That is XMO's infringement theory. Other categories of RTEs (categories 1, 2 and 4) identified by XMO are no longer at issue. Still, GD goes to great lengths to confuse the issues. GD re-casts XMO's theory as one depending on "dozens of intermediary files and disparate programs" to read information from a database. Resp. at 31; *see id*. at 29 ("XMO's expert relied on other files – not the accused RTE3 files – as reading from the database at issue."). That's not true. XMO never argued that "script.js" "reads information from a database" by calling on, in GD's words, a "pipeline of files." *See id*. at 7, 33, 38. GD concocted that theory as a strawman to blow down.

A careful review of the evidence GD cites reveals its misdirection:

GD cites Appx5882 ¶ 233 (Appx6013–6014) of Dr. Almeroth's Opening Report to argue that XMO claimed that "dozens" of files "form a pipeline to read information from databases." Resp. at 29. That is misleading and entirely inapplicable here. The cited paragraph concerned "Managed Word php page templates," a fourth category of RTEs used by Managed WordPress, not the RTE3 JavaScript file, "script.js," used by Website Builder and at issue here. Appx6009–

6014 (¶¶ 226–233).

GD cites Appx5268 ¶ 347 (Appx5430–5433), Appx5882 ¶¶ 229–231 (Appx6010–6013), and Dr. Almeroth's deposition transcript at 309:10–310:8 (Appx5850–5851), 310:23–311:25 (Appx5851–5852), and 318:7–320:11 (Appx5859–5861) to argue that "XMO's expert relied on other files – not the accused RTE3 files." Resp. at 29, 32, 37. This, too, is misleading and inapplicable. None of the cited evidence relates "script.js." Paragraphs 347 of Dr. Almeroth's Opening Report and paragraphs and 229–231 of his Reply Report concern the fourth category of RTEs "[f]or the Managed WordPress." Appx5430–5433; Appx6009–6014. And his deposition testimony is clearly traced to the first category of RTEs. *Compare* Appx5850–5861 (addressing ¶¶ 341 and 342 of Opening Report and ¶ 238 of Reply report) *and* Appx5426–5427 (¶¶ 341 and 342 of Opening Report, discussing RTE1 files,) *and* Appx6015 (¶ 238 of Reply Report, same).

GD cites Appx10569 ¶¶ 18–20 (Appx10577–10578) to argue that one of Dr. Almeroth's diagrams "showed the RTE3 files do not ever interact with (let alone 'read') the database." Resp. at 32. But Dr. Almeroth used those diagrams to illustrate how "script.js" is downloaded or "executed at runtime" by GD's customers to "display a webpage or website," other portions of the district court's claim construction of RTE not at issue here. Appx027; Appx10577. Those diagrams are unrelated to whether "script.js" "reads information from the database." *Id*. As to

this salient point, Dr. Almeroth explained in ¶ 20 of the same report, just below the cited diagrams, that the RTE3 "JavaScript files, such as script.js . . . read user selectable settings from the database using the function call fetch() to an API." Appx10578. GD conveniently ignores this.

Lastly, GD cites Appx5863 to claim that Dr. Almeroth admitted that "it is not actually the RTE3 files that are the 'mechanism by which data from [the accused database] is read'" but, instead, the mechanism is an API. Resp. at 37. Dr. Almeroth's testimony is entirely consistent with the point XMO is trying to make. "Databases have APIs on them," and they serve as an "interpreter" for any "get statement that reads from the database." Appx5863. In other words, as XMO and Dr. Almeroth have said all along, "fetch()" (of the "script.js" RTE3) itself "reads information from [GD's Cassandra]" database through DPS-API.

The rest of what GD cites is from GD's fact and expert witnesses, GD's legal briefs, and the district court's summary judgment opinion. *See* Resp. at 28–38. None of that bears on the true nature of XMO's RTE3 infringement theory, and the Court should not let GD distract from the issues that it must actually decide:

> What is the correct claim construction of the term, RTE, and, more specifically, does the construction include files that call on an API to retrieve information from a database?
>
> If the construction of RTE does include files that call on an API to retrieve information from a database, are there genuine disputes as to whether "script.js" (by way of "fetch()") calls on DPS-API to retrieve information from GD's Cassandra database?

The Court should answer both questions in the affirmative and reverse the district court's summary judgment of non-infringement.

B. The claim term, RTE, encompasses files that call on an API to retrieve information from a database.

The district court's construction of RTE has always covered files that read information from a database through an API. The only way the district court concluded on summary judgment that "script.js" is *not* an RTE was to exclude files that call on an API to retrieve information from a database. In this regard, this Court need only confirm that the plain language of the district court's construction ("read information from the database") does not exclude such files simply because they use an API. The Court should alternatively construe RTE to mean "file that is executed at runtime that *utilizes information from the database* and generates commands to display a web page or website," the construction that is best supported by the intrinsic evidence and one that, again, does not exclude files that call on an API.

1. The district court's construction of RTE, by its terms, includes files that "read information from a database" through an API.

The district court construed RTE as a file that "reads information from the database." At summary judgment, however, the district court interpreted its construction to exclude files that call on an API (with access codes to the database) to access and retrieve information from the database. Appx053–054. Nothing in the intrinsic evidence supports excluding such files. To the contrary, the claims broadly

recite that "run time files *utilize* information stored" in databases without limiting *how* that information is retrieved from the database (e.g., by using an API). Appx240–242 at 66:19–20, 69:1–2, 70:13–14; Appx344 at 65:3–6. The specification, moreover, discloses an RTE that relies on an API to read database information. Appx173 ("Runtime engine reads a param value which points to the database and initiates the read operation."); *see also* Appx5428 (XMO expert explaining that "downloading from a server with access to the database" is reading information from a database and is well known); Appx5434.

GD's experts agreed that RTE files use APIs to read information from databases. Dr. Greenspun testified that "using an API to access a database" is "the only way that – the only practical way that a Java program has to either *read from* or write to a database." Appx11118 at 115:8–15. Mr. Kent testified that reading information from a database "typically would be done through an API." Appx11540 at 169:8–13. And, when pressed to identify other ways an RTE can "read information from the database," Mr. Kent offered none, admitting that "*by definition* . . . you're going through some form of API." Appx11540 at 169:14–24; *see also* Appx11540–11541 at 169:8–13, 169:25–170:11.

GD declares that an API is not the only way to "read" a database, but it cannot credibly dispute its two experts. Resp. at 34. And XMO did not selectively cite "equivocal snippets" of their deposition transcripts. *Id*. at 35–36. The testimony

speaks for itself. Simply put, RTEs read information from databases by calling an API—that's how it works. So, if RTE is construed to be a file that "reads information from the database" (as the district court ruled), persons of ordinary skill in the art would understand that the file can do so by way of an API. There is no basis to require a file to "read information from a database," yet, at the same time, exclude the most common, "only practical way" that files "read information from a database."

The district court seemingly reached its conclusion by interpreting its claim construction to add the requirement of "direct" reading, that is, a file must "read[] information from the database *directly*" to be an RTE. Appx053. XMO maintains that the use an API to "read information from the database" (the langauge of the district court's construction) is a form of "direct" reading and not "indirect" reading. Dkt. 14, App. at 35–39. But, even if the use of an API is not "direct," GoDaddy offers no explanation of how the intrinsic evidence supports narrowing the plain terms of the district court's claim construction to exclude use of an API.

The Court should confirm that the use of an API is within the scope of the district court's claim construction.

2. The Court should alternatively construe RTE to require "utilizing information from the database."

The Court should alternatively adopt XMO's proposed construction, and construe RTE to mean a "file that is executed at runtime that *utilizes information*

from the database and generates commands to display a web page or website." This construction is grounded in the intrinsic evidence, which makes clear that the focus is *where* the RTE obtains information from (i.e., the database), not *how* the RTE obtains the information. Indeed, each claim of the '397 patent recites that "run time files *utilize* information stored" in databases. Appx240–242; *see also* App. § II.A.1.

Each of GD's arguments lacks merit. First, GD argues against a "facilitates retrieval" construction that XMO did not advance below and does not ask the Court to adopt here. Resp. 18–19, 20, 27. The Court should disregard those arguments.

Second, GD argues that XMO "should be estopped" from advancing its proposed "utilizes information" construction because XMO stated in a Patent Owner Response that the district court's construction of RTE is "consistent with *Phillips*." Resp. at 19 (citing *Facebook, Inc. v. Express Mobile, Inc.*, IPR2021-01126, Paper 19 at 18–20 (P.T.A.B. May 6, 2022)). GD omits important context. During claim construction, XMO objected that use of the term "read" rather than "utilize" might be used by GoDaddy to exclude embodiments where the RTE calls on an API to retrieve information from the database. *Supra* § I.B.1. After GoDaddy represented that it would not advance that argument, XMO accepted the district court's construction (Appx016–017) and later advanced that construction in the PTAB. It was only months later that the district court (now Judge Kennelly) interpreted the claim construction of RTE to exclude files that call on an API. Appx054. XMO

never "flip-flopped."

Third, GD admits that the claims of the '397 patent recite "run time file *utilizes* information stored in said database" (not "read, much less "*directly* read"), and that "run time file" includes RTE, but fails to explain why the plain claim language should not control. Resp. at 26. As to the '168 patent, GD misleadingly re-writes the claim language to fit its narrative. Claim 1 of the '168 patent recites that "runtime engine is configured to generate the web-site from the objects and style data *extracted from the provided database*." Appx344 at 65:4–6. The claim does *not* recite that the RTE is what extracted the data. Resp. at 26 (changing "extracted" to "extracting"). Rather, the claims are agnostic as to *how* the data is extracted from the database for use by the RTE. Appx344 at 65:4–5.

Fourth, GD does not dispute that the specification describes RTEs "utilizing" information retrieved from a database. *Compare* App. at 18–23 *and* Resp. at 20–21, 23–24. GD nonetheless argues that the specification demands a narrow construction. Resp. at 20–21. The specification does not provide a clear indication that XMO intended to limit RTE to require how the reading occurs. Moreover, nowhere does the specification limit "reading" to exclude the use of an API. Such narrow constructions are improper. *See Liebel–Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 913 (Fed. Cir. 2004).

Finally, GD argues that XMO characterized RTE as "reading and interpreting

the external database" during prosecution. Resp. at 21. But GD does not dispute that, despite making that statement, XMO later overcame the prior art by amending the claims to recite that RTEs "*utilize information*" from the database. Appx629–631; Appx665–666. The prosecution history, too, supports XMO's proposed construction.

C. There are genuine disputes of material fact that make summary judgment inappropriate.

"On appeal from a grant of summary judgment of non-infringement," this Court "determine[s] whether, after resolving factual inferences in favor of the patentee, the district court correctly concluded that no reasonable jury could find infringement." *Absolute Software, Inc. v. Stealth Signal, Inc.*, 659 F.3d 1121, 1129–30 (Fed. Cir. 2011). Here, the district court erred in granting summary judgment of non-infringement under either the district court's construction or XMO's proposed construction. The Court should reverse.

1. Under the district court's claim construction, there are genuine disputes as to whether "script.js" uses "fetch()" to "*read information from the database*."

There should be no doubt that "script.js" is an RTE under the district court's construction but, to the extent doubt remains, the Court must resolve inferences in XMO's favor. After doing so, there are at least genuine disputes of material fact that make summary judgment inappropriate.

a. There is a genuine dispute as to whether "script.js"

contains the function call, "fetch()."

There is, at the very least, a genuine dispute as to whether GD's "script.js"—the file that XMO identified as an RTE—contains the function call, "fetch()." After all, *Dr. Almeroth excerpted lines of code from "script.js"* and specifically highlighted "fetch()." Appx6006. GD, feigning ignorance, still argues that XMO provided only "wholly unsupported 'scintilla' of expert opinion that the RTE3 files even contained any of the *phantom 'fetch' calls* he proclaims to exist." Resp. at 31. That argument is not credible. GD knows about "fetch()." It is *recited verbatim in the code* for "script.js" that *Dr. Almeroth excerpted* and highlighted for GD. Excerpting lines of code is a far cry from "expert *ipse dixit*." *Id*. "script.js" clearly contains "fetch()," and regardless, on summary judgment, the Court must resolve all inferences in XMO's favor.

b. There is a genuine dispute as to whether "script.js" uses "fetch()" to "*read information from the database*."

There is a genuine dispute as to whether "script.js" (by way of its function call, "fetch()") "reads information from the database." Dr. Almeroth explained how "fetch()" requests information from GD's Cassandra database by calling to "DPS-API." Appx10577–10578. Every expert in this case agrees that calling on an API, like DPS-API, is how RTEs "read information from the database." Appx11118 at 115:8–15; Appx11540–11541 at 169:8–170:11. Dr. Almeroth then showed how "fetch()" returns that information to "script.js," which uses it to display a webpage.

Appx6005–6007; Appx5422. After resolving all factual inferences in favor of XMO, it would be reasonable for a jury to conclude that "script.js," in fact, "reads information from the database."

The district court erred because it improperly inferred (in GD's favor) that files other than "script.js" are used to "read information from the database." Appx053–055. There are no other files; only "script.js." As all experts agree, "DPS-API" does not break the chain between "script.js" and GD's Cassandra database because using an API is "the only practical way" a file can read information from a database. Appx11118 at 115:8–15; Appx5863 at 322:9–13.

What's left is what GD refers to as the "library of Priam files," but those are just database access codes that are part of "DPS-API." Appx5427; Appx4794–4796; *see also* Appx6018; Appx6021. GD admits that "Priam" is just a tool (specifically, a software driver, Appx5427 ¶ 343) used by "DPS-API" to access GD's Cassandra database. Appx4795 ¶ 35. There is no evidence (and GD points to none) establishing that Priam has any "fetching" function, as opposed to simply containing the codes to access the database.

The result is the same—"script.js" calls on "DPS-API" to retrieve information from the Cassandra database and, resolving all inferences in XMO's favor, it would be reasonable for a jury to conclude that "script.js" is an RTE.

c. There is a genuine dispute as to whether the information read by "script.js" (via "fetch()") comes

from GD's Cassandra database.

Lastly, there is a genuine dispute as to whether the information read by "script.js" (via "fetch()") comes from GD's Cassandra database, and not the third-party Content Delivery Network ("CDN"). Here, the district court found that "script.js" "interact[s] only with the CDN." Appx053. GD repeats this finding (Resp. at 9, 28), but does not meaningfully dispute that the district court got it wrong.

GD notes that its customers download "script.js" "from a third-party provider called a 'CDN.'" Resp. at 6 (citing Appx10578 ¶ 19). XMO does not dispute that fact, but it is not relevant to whether "script.js" is an RTE. What matters here is what happens after that download—the browsers of GD's customers execute "script.js" to trigger the "fetch()" function call and "read information from the database." Appx5418 ¶ 329; Appx5428 ¶¶ 344–345. This is what allows the browsers of GD's customers to dynamically build portions of web pages, and it is what makes "script.js" an RTE. Any other reference to the CDN is not relevant.

XMO came forward with evidence to specifically prove that "script.js," in fact, reads information from GD's Cassandra database and not some other third-party database. There is no dispute that (1) GD stores user-selected settings (e.g., colors, fontScale, etc.) on the Cassandra database (Appx5362–5363; Appx5988–5989; *see also* Appx5984–5993; Appx4794; Appx14764–14765 at 97:1–98:7; Appx17445 at 591:10–13); and (2) "DPS-API" is used to retrieve information from

the Cassandra database. Appx4795. Since there is a genuine dispute as to whether "script.js" "reads information from the database" by calling on "DPS-API" (*see supra* § I.B.1.), there is also a genuine dispute as to whether the information read by "script.js" comes from the Cassandra database itself.

Moreover, Dr. Almeroth showed (by excerpting lines of code) that executing "script.js" in a browser (thereby triggering "fetch()") causes "script.js" to dynamically populate with the same user-selected settings that GD admits are stored in its Cassandra database. Appx6005–6007; Appx5422. The reasonable inference, which should have been resolved in XMO's favor, is that the information read by "script.js" comes from GD's Cassandra database, not from the CDN or other, unidentified third-party database.

2. Under XMO's proposed claim construction, there is a genuine dispute as to whether "script.js" "*utilizes information from the database*."

There is also a genuine dispute as to whether "script.js" is an RTE under XMO's proposed construction. As described above in §§ I.B.1–2, resolving all inferences in XMO's favor, a reasonable jury could find that "script.js" utilizes information (user-selected settings) stored on GD's Cassandra database after that information is returned by "fetch()."

GD responds with one sentence. It argues that "there still is a failure of proof on this record that the 'fetch' call within the RTE3 files themselves 'utilize' any

resulting information actually flowing from the GoDaddy database." Resp. at 30. GD misses the point. XMO is not relying on "fetch()" as an RTE. "fetch()" is a function call within "script.js" used to retrieve information from the database—it is "script.js" that "utilizes information from the database" after retrieval. GD provides no response to this, and, on summary judgment, any factual inferences required must be resolved in XMO's favor.

II. '755 PATENT FAMILY: THE DISTRICT COURT ERRED IN DENYING XMO'S MOTION FOR A NEW TRIAL AND JUDGMENT AS A MATTER OF LAW

The district court erred in denying XMO's motion for a new trial and JMOL as to the '755 Patent Family. GD contradicted the district court's construction of "Player" and the impact cannot be overstated. GD prejudiced XMO's ability to fairly present its infringement case and infected the verdict by inviting the jury to assign incorrect meanings to critical claim terms. The Court should reverse.

A. XMO was entitled to a new trial because GD contradicted the construction of "device-specific code."

The asserted '755 Patent Family claims recite a "Player" and an "Application." *See* App. at 42. "Player" means "device-specific code which contains instructions of a device and which is separate from the" claimed "Application." Appx037; *Shopify Inc. v. Express Mobile, Inc.*, CA No. 19-439-RGA, 2020 WL 3432531, at *7 (D. Del. June 23, 2020) (hereinafter "*Shopify Claim Construction*" available at Appx1852). "Device-specific code"—and the equivalent

term "device-dependent code"—mean "code that is specific to the operating system, programming language, or platform of a device." Appx010; Appx1852. At trial, GD's expert improperly contradicted those constructions.

GD devotes less than a page weakly opposing XMO's new trial request. Resp. 57–58. GD entirely fails to address pertinent decisions of this Court and XMO's substantive arguments that GD's repeated and egregious contradiction of the district court's construction is exactly the type of case that demands a new trial.

1. GD's expert improperly equated "device-specific code" with code specific to an "operating system."

"Device-specific code" means code that is somehow specialized to one of *three different aspects* of a device: its "operating system," its "programming language," or its "platform." Appx010. Under that construction, code that is specialized for a device's particular "platform" would be device-specific, even if it is not specialized for a particular "operating system." GD's expert Mr. Kent, however, rewrote that construction at trial. While he acknowledged that the construction refers to a device's "programming language, operating system, *or platform*," he insisted that "*platform*" "essentially" just refers to the "*operating system*." Appx17454 at 628:13; *see* Appx17467 at 679:12–14 ("Q. And so you're defining *device dependent* as only operable on a single type of device, correct? A. Yeah, obviously, by *device* it means *operating system*."); Appx17469 at 686:15–17; *see also* Resp. at 43–45. For example, Mr. Kent presented a demonstrative of code

specialized to "different operating systems," and insisted it illustrated the court's "construction" of the device-specific "player." Appx17472 at 698:5–10.

Consistent with that view, Mr. Kent characterized the term "platform" in the court's construction as a superfluity. According to him, "platform" just refers "to the operating system running on [a] piece of hardware." Appx17472 at 699:7–19; *see id*. at 699:20–22; Appx17472 at 699:24–700:3. Mr. Kent's effort to rewrite the court's claim construction was improper and required a new trial. *See Exergen Corp. v. Wal-Mart Stores, Inc.*, 575 F.3d 1312, 1321 (Fed. Cir. 2009); *Moba, B.V. v. Diamond Automation, Inc.*, 325 F.3d 1306, 1313-14 (Fed. Cir. 2003); Resp. at 41–43.

Each of GD's efforts to rehabilitate these improper and persistent errors fails.

First, instead of showing how those statements by Mr. Kent were proper, GD, like the district court, points to other testimony where Mr. Kent "repeated the court's claim construction" in full. Resp. at 41; *see* Appx128–130. But this Court's precedent prohibits experts from offering "misleading statement[s]" that contradict the court's claim construction. *Exergen*, 575 F.3d at 1321. Misleading statements are not made acceptable by providing proper testimony elsewhere.

Mr. Kent's other testimony did nothing to cure his repeated misstatements. Mr. Kent did not merely omit the word "platform" on certain occasions when addressing "device-specific code." Instead, Mr. Kent testified that the court's

construction "essentially" just boiled down to the "operating system." Appx17454 at 628:13. And Mr. Kent told the jury that the term "platform"—a distinct part of the construction, separate from "operating system"—was essentially synonymous with "operating system." According to Mr. Kent, "platform" refers to a "device running a particular operating system." Appx17472 at 699:20–22; *see id.* at 699:15–16; Appx17472 at 700:3. Mr. Kent's testimony conveyed to the jury that the three parts of the construction—"operating system, programming language, or platform"—were really just one: "operating system."

GD, like the district court, fails to explain how Mr. Kent's other testimony mitigates the misimpression from his improper testimony. *See* Resp. at 40–59; Appx128–130. GD does not point, for example, to anywhere that Mr. Kent corrected himself by explaining that the court's construction encompassed three different things, of which "operating system" was just one. Nor does it point to any testimony where Mr. Kent corrected himself by noting that "platform" could refer to aspects of the device *other than* the operating system.

GD attempts to invoke *Verizon Servs. Corp. v. Cox Fibernet Va., Inc.*, 602 F.3d 1325, 1335 (Fed. Cir. 2010). *See* Resp. at 41–42. There, this Court rejected a new trial because defendant's expert, on cross-examination, expressly disavowed a prior suggestion that a construction was limited when it was not so limited in reality. *Verizon*, 602 F.3d at 1334. GD identifies no similar curative testimony from Mr.

Kent. To the contrary, unlike the expert in *Verizon*, Mr. Kent doubled-down on his mischaracterization during cross-examination. Appx17467 at 678:14–679:14. Mr. Kent could not have been clearer that his non-infringement opinion was based on "defining" the term "device dependent" to mean: "only operable on a single type of … operating system." *Id.* The jury was not entitled to find non-infringement based on that basis. *See WesternGeco LLC v. ION Geophysical Corp.*, 913 F.3d 1067, 1073 (Fed. Cir. 2019). A new trial is required.

Second, GD, like the district court, accuses XMO of raising an "impermissible post-trial claim construction" issue. Resp. at 42–43; *see* Appx129–130. Not so. As explained previously and below, GD's admitted effort to exclude different web browsers from the scope of the term "platform" contradicts the claim constructions in this case and provides a separate reason why a new trial was required. *See infra* § II.A.2.; App. § III.A.1. But Mr. Kent's testimony contradicted the claim construction even putting aside the meaning of the term "platform." The district court's construction—based on the patent's express definition—states that "device-specific code" is code specialized for a particular "operating system, programming language, *or platform*." Appx010. Code specialized to *any* of those three aspects of the device can be "device-specific" (or "device-dependent") code. It is axiomatic that those three terms—"operating system," "programming language," and "platform"—must have different meanings. *See Tandon Corp. v. ITC*, 831 F.2d

1017, 1023 (Fed. Cir. 1987). Far from seeking further construction, XMO simply asked the district court to enforce the distinction between those three aspects inherent in the plain and ordinary meaning of the court's construction. App. at 50–53; *contra* Resp. at 44–45.

For the same reason, XMO's request for a new trial did not seek to "'purge every shred of ambiguity'" from the court's construction. Resp. at 43–44. There is no ambiguity. The problem is that GD's expert tried to collapse the three possibilities of the construction into one. There is nothing unreasonable—and no possibility of claim construction "'proceed[ing] *ad infinitum*'"—in requesting the district court to enforce the inherent plain meaning of the construction it already adopted.

Third, GD fails in its attempt to argue that XMO somehow "waived" its objection to Mr. Kent's testimony. Resp. at 47–49. GD raised that argument before the district court. Appx21735–21736. The district court did not find waiver, and instead addressed XMO's arguments on the merits. Appx127–130. It is well established that "a party may raise on appeal any issue that was … actually decided below." *Lifestyle Enters., Inc. v. United States*, 751 F.3d 1371, 1377 (Fed. Cir. 2014).

GD acknowledges that "XMO filed a motion in limine on this very issue," but suggests the district court somehow left the issue open for trial. Resp. at 48–49. To

the contrary, the district court issued a definitive ruling, granting XMO's motion "as a no-brainer." Appx17001 at 49:7–14. And while GD invokes cases holding that parties may not "present new claim construction disputes if they are raised for the first time after trial," Resp. at 49, XMO does not seek a new claim construction here, *see supra*.

2. GD's expert also improperly excluded web browsers from the scope of the term "platform."

At trial, GD's expert Mr. Kent did not dispute the opinion of XMO's expert Dr. Almeroth that the accused system contains code that detects browsers and alters the "look and feel" of the system based on the type of "browser." Appx17468 at 684:7–18. Dr. Almeroth testified that a browser is a type of "platform," and therefore browser-specialized code is a type of device-dependent code. Appx17368–17369 at 284:19–286:3. Instead of disputing that the accused GD system contains browser-specialized code, Mr. Kent "[dis]agree[d] with [Almeroth's] definition" of the term "device dependent." Appx17468 at 684:19–23. According to Mr. Kent, a browser cannot be a type of "platform" within the scope of the court's construction. Appx17467 at 680:2–21. That testimony, too, contradicted the prior rulings in this case and the related *Shopify* case, and necessitated a new trial on infringement. App. at 42–49.

First, GD does not deny that the construction of "device-specific code" in this case was taken from the construction in the *Shopify* case. GD also does not deny

that the *Shopify* court understood a "browser" to be a type of "platform." Resp. at 42; *see* App. at 42–44. And finally, GD does not deny that Mr. Kent expressly testified that a browser *is not* a type of platform. Instead, GD insists that its expert was entitled to depart from those rulings at trial. That position overlooks the procedural history of this case. Both parties relied heavily on the prior *Shopify* constructions. In particular, they agreed to import the *Shopify* constructions of "device-dependent"/"device-specific" code. Appx1720. GD never argued during claim construction that "device-specific code" and "device-dependent code" should be given different constructions than in the *Shopify* case.

Then, in September 2021, Judge Andrews—who still had both the *GoDaddy* and *Shopify* cases before him—issued the summary-judgment order in *Shopify*. There, the defendant urged that code containing "conditional logic" specialized to a particular "*browser version and type*," is not "device-dependent code." Appx6508. Judge Andrews rejected that argument, accepting XMO's view that "conditional JavaScript code" specialized to different browsers is "device-specific code because a browser is understood in the art as a platform (and therefore meets [the court's] construction for 'device-dependent code." *Id.*

When the parties filed their summary judgment briefs in November 2021—months after the *Shopify* summary-judgment ruling, when both cases were still before Judge Andrews—GD still did not argue that browser-specialized code could

not be device-specific code. Appx5244–5247. To the contrary, GD "[did] not dispute that the JavaScript Player code" in its system "contains device-dependent code." Appx056. After briefing was complete, the *GoDaddy* case was reassigned to Judge Kennelly, who rejected GD's argument, finding that "this issue was addressed in *Shopify*," and "*adopt[ing] the same reasoning as in that case*." *Id*. Judge Kennelly's opinion gave no indication that he endorsed less than the full reasoning from the *Shopify* case, or departed from Judge Andrews's understanding of whether browser-specialized code met the requirement of code specific to a "platform." Thus, GD, contrary to its arguments in Resp. at 42, was not entitled to argue for a different meaning of "platform" at trial. That issue was resolved at summary judgment, by "adopt[ion]" of the reasoning in *Shopify*.

GD invokes the truism that the district court was not "bound by" the orders and claim constructions in *Shopify*. Resp. at 42. But statements by the district court that it was not required to follow the *Shopify* rulings *in general* are irrelevant and fail to establish that the district court did not adopt the applicable *Shopify* ruling here. The *Shopify* court's conclusion that browsers are a type of platform is accordingly binding in this case. And that is why it was an error for the district court to adopt a contrary view below. *See Wi-LAN, Inc. v. Ericsson, Inc.*, 675 F. App'x 984, 994 (Fed. Cir. 2017); App. at 51–52.

Second, GD's, and the district court's, assertion that XMO is belatedly seeking a construction of "platform" to include "browsers" fails for the same reason. Resp. at 42–43; *see* Appx129–130. XMO had no reason to do so. That was already in the construction.

B. The district court also erred in denying XMO's motion for JMOL.

The district court also erred in denying XMO's motion for JMOL. GD presented no legally sufficient evidence for a reasonable jury to find non-infringement.

First, for the reasons set forth above as to "Player," GD infected the verdict by consistently and improperly presenting a contradictory claim construction. JMOL is warranted under these circumstances. *See Moba, B.V.*, 325 F.3d at 1313–1314 (reversing denial of JMOL where verdict was based on arguments that contradicted claim construction); *Eon Corp. IP Holdings v. Silver Spring Networks*, 815 F.3d 1314, 1316 (Fed. Cir. 2016) (same); *NobelBiz, Inc. v. Global Connect, L.L.C.*, 701 F. App'x 994 (Fed. Cir. 2017) (same).

GD also argues that Dr. Almeroth's testimony "dooms" infringement because the accused player contains portions that are not device-dependent and those portions receive the web service output. Resp. at 52–53. Not so. As GD quotes in its brief, Dr. Almeroth testified that the accused player was present, regardless of particular code segments within what he identified as the player not being device-

dependent. *Id*. (quoting Appx17534 at 945:6–15). And Dr. Almeroth also clarified that the particular code segments that GD myopically focuses on were only "part" of the code that receives the output, noting that "[t]hese files are very long. I'm not showing everything in the file." Appx17533 at 943:24–944:7.

Second, GD infected the verdict as to the "registry" limitation. Here, GD presented a contradictory claim construction for the term "registry," inviting the jury to incorrectly believe that a "database" needs to be a structured database, rather than an unstructured database. *Compare* App. at 56–57 *and* Resp. at 53–56. GD cites testimony of its expert Mr. Kent at Appx17460, but conveniently omits Mr. Kent's testimony immediately preceding his Post-it note analogy: "So a database is effectively, it's a collection of structured data. . . . It has to be structured." This is XMO's point—GD presented evidence that contradicted the district court's claim construction of "registry" because nothing in the construction narrows "database" to a "structured database." From what's left in the record, there is no basis for a reasonable jury to find that GD's databases – structured or unstructured – are not databases.

Lastly, as to the "symbolic names" limitation, GD concedes that its trial presentation concerned features (e.g., YouTube) that XMO did not identify as a "symbolic name" at trial (e.g., "Contact Form" and "Add to Cart"). *Compare* App. at 57 *and* Resp. at 56–57. GD says that it "rebutted XMO's evidence as to the

presence of Symbolic Names" (*id*. at 57), but only GD's citations to Appx17437 and Appx17443 even tangentially relate the features identified by XMO. And, even then, the testimony cited by GD about "div IDs" is non-responsive because XMO never relied on the randomly-generated "div IDs" as symbolic names. Appx17524; Appx17256. GD's arguments that any player does not receive web service output values fail for the same reason. Resp. at 51 (again referencing YouTube). The evidence that XMO actually presented to establish that GD's systems practice the "symbolic names" limitation and that the accused player receives that symbolic names output went unrebutted. And, on this record, no reasonable jury could have found non-infringement based on the absence of "symbolic names."

III. THE COURT SHOULD DISREGARD GD'S ARGUMENT ABOUT WILLFUL INFRINGEMENT

The Court should disregard GD's request for an advisory opinion to resolve "a split in the Delaware District as to willful infringement." Resp. at 58. The district court denied summary judgment of no willfulness as to the '397 Patent Family, but granted summary judgment of non-infringement. Appx053–055, Appx057–058. As to the '755 Patent Family, the jury returned a verdict of both non-infringement and no willful infringement. Appx114. GD is the prevailing party and its arguments do not advance alternative bases to support the judgment of non-infringement as to either patent family. *See Aug. Tech. Corp. v. Camtek, Ltd.*, 542 F. App'x 985, 994 (Fed. Cir. 2013).

March 1, 2024

/s/ James R. Nuttall

James R. Nuttall
Katherine H. Tellez
Robert F. Kappers
Candice J. Kwark
STEPTOE LLP
227 West Monroe Street, Suite 4700
Chicago, IL 60606
Telephone: (312) 577-1300
Facsimile: (312) 577-1370

Counsel for Plaintiff-Appellant, Express Mobile, Inc.

CERTIFICATE OF COMPLIANCE

I hereby certify that this brief complies with the type-volume limitations of FED. CIR. R. 32(A). This brief contains 6,930 words (including diagrams and images), excluding the parts of the brief exempted by FED. R. APP. P. 32(F) and FED. CIR. R. 32(B), as counted by Microsoft® Word 2010, the word processing software used to prepare this brief.

This brief complies with the typeface requirements of FED. R. APP. P. 32(A)(5) and the type style requirements of FED. R. APP. P. 32(A)(6). This brief has been prepared in a proportionally spaced typeface using Microsoft® Word 2010, Times New Roman, 14 point.

Dated: March 1, 2024

/s/ James R. Nuttall

James R. Nuttall
(Lead Counsel)
jnuttall@steptoe.com
Katherine H. Tellez
Robert F. Kappers
Candice J. Kwark
STEPTOE LLP
227 West Monroe Street, Suite 4700
Chicago, IL 60606
Telephone: (312) 577-1300
Facsimile: (312) 577-1370

Counsel for Plaintiff-Appellant, Express Mobile, Inc.

CERTIFICATE OF SERVICE

I hereby certify that, on the 1st day of March, 2024, I electronically filed the foregoing with the Clerk of Court using the CM/ECF system which thereby served a copy upon all counsel of record.

Upon acceptance by the Court of the e-filed document, six paper copies of the brief will be filed with the Court via Federal Express, priority overnight, within the time provided in the Court's rules.

March 1, 2024

/s/ James R. Nuttall

James R. Nuttall
(Lead Counsel)
jnuttall@steptoe.com
Katherine H. Tellez
Robert F. Kappers
Candice J. Kwark
STEPTOE LLP
227 West Monroe Street, Suite 4700
Chicago, IL 60606
Telephone: (312) 577-1300
Facsimile: (312) 577-1370

Counsel for Plaintiff-Appellant, Express Mobile, Inc.